WHAT IS CLAIMED IS:

1. A method for processing information comprising the steps of:

providing a frame key based on a master key and a frame number of a frame of information to a keystream generator as an initial fill for one or more registers of the keystream generator such that the initial fill establishes a state for the one or more registers of the keystream generator; and

generating, at the keystream generator, a keystream sequence based on the state established by the initial fill, wherein the keystream sequence includes a modulo-2 sum of a lagged-Fibonacci sequence and a pseudo-random sequence produced by a filter generator.

2. The method of claim 1, further comprising the step of:

creating a second frame key based the master key and a second frame number such that the second frame key represents a second frame of information.

3. The method of claim 1, wherein said providing further comprises the step of:

initializing the one or more registers of the keystream generator based on the frame key.

4. The method of claim 1, wherein said generating further comprises the step of:

shifting one or more bits of a first word corresponding to one of a plurality of registers of the keystream generator; and shifting one or more bits of a second word corresponding to a second of the plurality of registers.

5. The method of claim 4, further comprising the step of:

combining the shifted first word with the shifted second word based on exclusive OR logic.

6. The method of claim 3, wherein said initializing further comprises the step of:

initializing a first linear feedback shift register of the filter generator with the frame key.

7. The method of claim 6, further comprising the step of:

defining the first linear feedback generator of the filter generator as producing the pseudo-random sequence that satisfies the following equation:

$S_n=S_{n-2} \oplus S_{n-3}<<31 \oplus S_{n-4}>>1, n\geq4,$

wherein << corresponds to a zero-fill left-shift operation, >> corresponds to a zero-fill right shift operation, ⊕ denotes XOR logic, and n is the number of stages in the first linear feedback shift register.

8. The method of claim 1, wherein said generating further comprises the step of:

producing the lagged-Fibonacci sequence using a lagged-Fibonacci generator.

9. The method of claim 8, further comprising the step of:

initializing a second linear feedback shift register of the lagged-Fibonacci generator with the frame key.

10. The method of claim 8, further comprising the step of:

defining the lagged-Fibonacci sequence as satisfying the following equation:

$$L_n=L_{n-5}+L_{n-17} \bmod 2^{32} ,$$

wherein n corresponds to the number of stages of in the second linear feedback shift register and is greater than or equal to 17 words of 32-bits, and mod 2^{32} corresponds to modulo-2 addition of a 32 bit word.

11. The method of claim 8, further comprising the step of:

rotating an output of the lagged-Fibonacci generator.

12. The method of claim 1, further comprising the step of:

producing a bit stream of ciphertext by modulo-2 adding each bit in the keystream sequence produced by the keystream generator with a corresponding bit in a bit stream of plaintext.

13. The method of claim 1, further comprising the step of:

producing a bit stream of plaintext by modulo-2 adding each bit in the keystream sequence produced by the keystream generator with a corresponding bit in a bit stream of ciphertext.

14. A system for processing information, said system comprising:

at least one memory including:

code that provides a frame key based on a master key and a frame number of a frame of information to a keystream generator as an initial fill for one or more registers of the keystream generator such that the initial fill establishes a state for the one or more registers of the keystream generator, and

code that generates, at the keystream generator, a keystream sequence based on the state established by the initial fill, wherein the keystream sequence includes a modulo-2 sum of a lagged-Fibonacci sequence and a pseudo-random sequence produced by a filter generator; and

at least one processor that executes said code.

15. The system of claim 14, further comprising:

code that creates a second frame key based the master key and a second frame number such that the second frame key represents a second frame of information to be enciphered.

16. The system of claim 14, wherein said code that provides the frame key further comprises:

code that initializes the one or more registers of the keystream generator based on the frame key.

17. The system of claim 14, wherein said code that generates further comprises:

code that shifts one or more bits of a first word corresponding to one of a plurality of registers of the keystream generator; and

code that shifts one or more bits of a second word corresponding to a second of the plurality of registers.

18. The system of claim 17, further comprising:

code that combines the shifted first word with the shifted second word based on exclusive OR logic.

19. The system of claim 16, wherein said code that initializes the one or more registers further comprises:

code that initializes a first linear feedback shift register of the filter generator with the frame key.

20. The system of claim 19, further comprising:

code that defines the first linear feedback generator of the filter generator as producing the pseudo-random sequence that satisfies the following equation:

$$S_n=S_{n-2} \oplus S_{n-3}<<31 \oplus S_{n-4}>>1, n\geq 4,$$

wherein << corresponds to a zero-fill left-shift operation, >> corresponds to a zero-fill right shift operation, ⊕ denotes XOR logic, and n is the number of stages in the first linear feedback shift register.

21. The system of claim 14, wherein said code that generates a keystream sequence further comprises:

code that produces the lagged-Fibonacci sequence using a lagged-Fibonacci generator.

22. The system of claim 21, further comprising:

code that initializes a second linear feedback shift register of the lagged-Fibonacci generator with the frame key.

23. The system of claim 21, further comprising:

code that defines the lagged-Fibonacci sequence as satisfying the following equation:

$$L_n=L_{n-5}+L_{n-17} \bmod 2^{32} ,$$

wherein n corresponds to the number of stages of in the second linear feedback shift register and is greater than or equal to 17 words of 32-bits, and mod 2^{32} corresponds to modulo-2 addition of a 32 bit word.

24. The system of claim 21, further comprising:

code that rotates an output of the lagged-Fibonacci generator.

25. The system of claim 14, further comprising:

code that produces a bit stream of ciphertext by modulo-2 adding each bit in the keystream sequence produced by the keystream generator with a corresponding bit in a bit stream of plaintext.

26. The system of claim 14, further comprising:

code that produces a bit stream of plaintext by modulo-2 adding each bit in the keystream sequence produced by the keystream generator with a corresponding bit in a bit stream of ciphertext.

27. A computer program product, the computer program product comprising code, said code comprising:

code that provides a frame key based on a master key and a frame number of a frame of information to a keystream generator as an initial fill for one or more registers of the keystream generator such that the initial fill establishes a state for the one or more registers of the keystream generator; and code that generates, at the keystream generator, a keystream sequence based on the state established by the initial fill, wherein the keystream sequence includes a modulo-2 sum of a lagged-Fibonacci sequence and a pseudo-random sequence produced by a filter generator.

28. The computer program product of claim 27, said code further comprising:

code that produces a bit stream of ciphertext by modulo-2 adding each bit in the keystream sequence produced by the keystream generator with a corresponding bit in a bit stream of plaintext.

29. The computer program product of claim 27, said code further comprising:

code that produces a bit stream of plaintext by modulo-2 adding each bit in the keystream sequence produced by the keystream generator with a corresponding bit in a bit stream of ciphertext.

30. A hand held device for communicating information, said hand held device comprising:

at least one memory including:

code that provides a frame key based on a master key and a frame number of a frame of information to a keystream generator as an initial fill for one or more registers of the keystream generator such that the initial fill establishes a state for the one or more registers of the keystream generator, and code that generates, at the keystream generator, a keystream sequence based on the state established by the initial fill, wherein the keystream sequence includes a modulo-2 sum of a lagged-Fibonacci sequence and a pseudo-random sequence produced by a filter generator;

at least one processor that executes said code.

31. The hand held device of claim 30, further comprising:

code that receives code and information from a base station.

32. The hand held device of claim 31, further comprising:

code that configures the keystream generator based on the received code and information.

33. The hand held device of claim 30, further comprising:

code that receives one or more keys from a base station such that the one or more keys initialize one or more registers of the keystream generator.